REMARKS

1. Claims 1-4, 20-23, and 36-42 were and are now pending. No claims have been amended or cancelled. A clean copy of the presently pending claims is now of record as set forth in Amendment B to the captioned patent application that was mailed on January 7, 2002 (herein after, **Amendment B**). Reexamination and reconsideration of the application are requested.

2. Rejections under 35 U.S.C. 102(e) and 103(a)

Claims 1-4 and 36 were rejected in the Office Action under pre-AIPA 35 U.S.C. 102(e) as being anticipated by Haff et al. (US Patent No. 6,219,669). Claims 20-23 and 37-42 were rejected in the Office Action under 35 U.S.C. 103(a) as being unpatentable over Haff et al. (US Patent No. 6,219,669) in view of Kolling et al. (US Patent No. 5,963,925). The Applicant respectfully traverses the rejections and requests consideration of the following.

3. Data vs. File

The term "data" can be distinguish from the term "file". A "file" is a collection of data or information that has a name, called the filename. Almost all information stored in a computer must be in a file. There are many different types of files: data files, text files, program files, directory files, and so on. A data file is not limited to contain only one particular type of data, but rather can contain many different particular types of data. In contrast, "data" is distinct pieces of information, usually formatted in a special way. All software is divided into two general categories: data and programs. Programs are

collections of instructions for manipulating data.

4. Haff et al. Teach File Packets Particularized by Files, Not Data

Haff et al. teach at Col. 22, lines 10-16 a user interface by which a user can make a selection, by a "drag and drop feature", of certain files from all files that are displayed on the user interface. While Haff et al. do not teach that all of the selected files contain the same particular type of data or that each of the selected files contains many different particular types of data, Haff et al. do teach that the files that are selected by the user are those files that the user wishes to transmit.

Haff et al. teach that the user-selected files are formed into a "file packet" that contains the user-selected files. In contrast, Haff et al. do not teach that file packet is formed so as to be particularized to contain and carry a particular type of data. As such, Haff et al. do not teach, suggest, or imply that a file packet is to contain any particular type of data, but rather is to contain only those particular files that were selected by a user without *any* limitation or requirement as to the type of data that is in each file *or* as to the type of data in each file packet. Accordingly, Haff et al. disclose a file packet based on file selection, not data selection.

5. Applicants Disclose Parcel Components Particularized by Data, Not Files

For a better understanding of the present invention, we review Figure 5 and its description in the specification, at page 18, lines 4-21, as follows:

> The BIS gateway 80 has a parcel manager 134 to transfer billing data and other information from the BIS to the service center. The parcel manager transfers the data in "parcels". The parcel manager 134 is responsible for reliably transferring parcels from the BIS 34 to the service center and tracking the parcels as they go from computer to computer. It

> is this tracking function that enables the management console UI 100 to show the location and status of particular parcels. The parcel manager 134 is described below in more detail with reference to Fig. 7.
>
> Atop the parcel manager 134 are a set of handlers that collectively form an enterprise interface into the parcel manager. The interface handlers handle ***requests to create different types of parcels, depending upon the type of information being transferred*** to the service center. The enterprise interface handlers include a consumer information handler 136, a payment handler 138, a batch handler 140, and a template handler 142. The handlers facilitate ***creation of particularized parcels for shipment*** to the service center. For instance, the batch handler 140 facilitates creation of statement batch parcel to be transferred to the service center. The handlers 136-142 are preferably implemented as COM (component object model) objects and are called via a set of enterprise integration APIs. (emphasis added)

The foregoing text presents the concept of particularization of the parcel contents. As designed, each parcel is to be limited as to its content. This limitation placed upon the content of the parcel is dictated by a request. The request is for a particular data type and is directed to a particular type of interface handler. The particular interface handler to which a particular request is directed is responsible for the requested particular type data in the particular type of parcel. As such, the Applicant provides for a particular type of interface handler to assemble a particular type of parcel to be made up of a particular type of data. The application provides for at least four (4) particular types of data: consumer information, payment, statement batch, and template.

The Applicant respectfully submits that the assembly of a file packet from user selected files taught by Haff et al. is nonanalgous to the recited parcel limitation in each of the independent claims. Moreover, the parcel limitation of the claimed invention is particular as to the type of data. This limitation is further narrowed to be *recited as being selected from the group consisting of consumer information data, payment data, batch statement data, and statement template data.*

6. **Comments on Kolling et al.**

The comments expressed in **Amendment B** with respect to Kolling et al. and the pending claims are incorporated herein by reference.

7. **Additional Comments**

Neither Haff et al. nor Kolling et al. teach, suggest, or imply, either alone or in the combination, a file packet that is particularized or otherwise restricted as to its contents by a particular type of data. The present specification proposes to construct parcels of data, where each parcel can be limited to one of at least four different kinds of data. Inherent benefits are realized from this concept. In addition to the direct benefit of the reliable transfer of specifically requested parcels as they go in a network from one computer to another between the biller and the service center, the data traffic on the network is not congested by unrequested data. Network congestion is avoided by the claimed invention in that the data in each parcel is particularly limited to that data that was specifically requested.

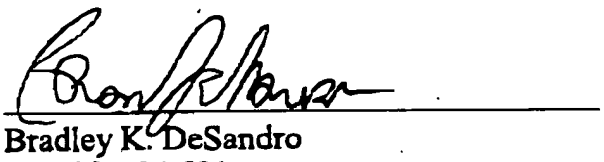
Neither Haff et al. nor Kolling et al. teach particularly requested and composed parcels of data. Since the file packets of Haff et al. are not limited to be particularized to contain and carry a particular type of data that was requested, Haff et al. do not achieve this benefit of reduced network congestion. Neither do the electronic statement presentment systems taught by Kolling et al. achieve this benefit of reduced network congestion.

8. Conclusion

In sum, neither Haff et al. nor Kolling et al. teach, suggest, or imply, either alone or together, the combinations of the recited elements in the pending independent Claims 1-4. The Applicant respectfully submits that pending Claims 1-4 and 36 are not anticipated by Haff et al. and that, with respect to pending Claims 20-23 and 37-42, a *prima facie* case of obvious has not been made out. As such, the Applicant respectfully maintains that the pending independent Claims 1-4 are allowable, as are the claims respectively depending therefrom. Accordingly, the present application is in condition for allowance. Reconsideration of the rejections is requested. Allowance of Claims 1-4, 20-23, and 36-42 at an early date is solicited.

Respectfully Submitted,
Lee & Hayes PLLC

Dated: 7/26/02

By: 

Bradley K. DeSandro
Reg. No. 34,521
(509) 324-9256, Ext. 228

Specification Amendment Mark up under 37 C.F.R. § 1.121(b)1(iii)

A. In accordance with 37 C.F.R. § 1.121(b)1(iii), a marked up version of the amended paragraph in the specification at Page 8, lines 14 through 26 is submitted by this separate paper as follows:

The template is preferably constructed using as Active Server Pages, a technology introduced by Microsoft Corporation. An active server page, or "ASP", allows a user to define templates using a combination of a hypertext language (e.g., HTML) and a scripting language, such as Visual Basic Script (or "VBS") or JScript from Microsoft Corporation, perl, python, REXX, or tcl. The HTML language defines the basic structure of the billing statement and the scripting language defines which data is inserted into the appropriate fields. The scripting instructions are set apart by special delimiters. When an ASP file is read and rendered, the scripting instructions within the delimiters are executed to fill in the billing data. The result is a billing statement in a pure hypertext document. Active Server Pages are described in documentation available from Microsoft Corporation of Redmond, WA, USA.[’s Web site "www.microsoft.com", under the section Internet Information Services. This text is hereby incorporated by reference.]

B. In accordance with 37 C.F.R. § 1.121(b)1(iii), a marked up version of the amended paragraph in the specification at Page 26, line 17 through Page 27, line 2 is submitted by this separate paper as follows:

Fig. 7 shows the BIS parcel manager 134 in more detail. Applications 220 running at the biller computer system use the parcel manager 134 to create a parcel, send the parcel across to a computer at the service center, and receive notifications on the status and location of the parcel as it moves from one machine to another. Applications 22[0]0 interface with the parcel manager 134 via the APIs in the enterprise interface 222, which consists of the consumer information handler 136, the payment handler 138, the batch handler 140, and the template handler 142 (see Fig. 5). The management console 98 works with the parcel manager 134 to track the parcels between computers. It is noted that the parcel manager 154 residing at the service center gateway 86 is essentially the same, and is not described in detail.

C. In accordance with 37 C.F.R. § 1.121(b)1(iii), a marked up version of the amended paragraph in the specification at Page 20, line 18 through Page 21, line 2 is submitted by this separate paper as follows:

The BIS 34 is implemented as software modules stored in program memory 192. The modules—billing data translator module 2[8]7, statement designer module 62, rules manager module 66, resource manager module 70, and advertising manager module 74, management console module 98, accounts receivable translator module 94, payment translator module, and gateway 80—run on the operating system. In a preferred implementation, the resource manager 70 and advertising manager 74 are implemented as HTML development software, such as Visual InterDev from Microsoft Corporation. The statement designer 62 and the rules manager 66 are implemented as extensions of the Visual InterDev software. The billing data 60, templates 64, rules 68, resources 72, advertising information 76, and payment/remittance information 92 are stored in the data memory 186.

D. In accordance with 37 C.F.R. § 1.121(b)1(iii), a marked up version of the amended paragraphs in the specification at Page 49, line 23 through Page 50, line 19 are submitted by this separate paper as follows:

Exemplary Task 2: Fig. 10 shows a method for handling a batch of billing data for an installed template. The biller creates billing data using its legacy billing system. The billing data is passed through the statement data translator 2[8]7 (step 290). The translator instantiates a statement batch object to hold the data (step 292). The translator 2[8]7 specifies the biller and the template to be associated with the billing data (step 294) and validates the specified biller and template against records of authorized billers and installed templates received from the service center (step 296). This validation process ensures that the billing data is for an approved biller recognized by the service center and is for a template that is installed at the service center. The statement translator 2[8]7 then loads data into the statement batch object. The statement batch object accepts data that complies with the available fields in the industry schema tables.

The BIS gateway assigns a batch ID to the statement batch and a statement ID to each statement in the batch (step 298). The statement data translator 2[8]7 calls via the batch handler 140 into the parcel manager interface 224 to create a statement batch parcel (step 300). The batch parcel contains the following information: biller ID, batch ID, template ID, template rule ID, resource table records, statement table records, and industry table records. The batch parcel is sent to the service center during the next connection with the service center (step 302). The service center processes the batch parcel and loads the data into the service center

database (step 304). The service center's parcel manager generates and returns a bulletin indicating that the batch has been received and loaded at the service center (step 306).